Programming: A Subset of Problem Solving

Francis Sommerville

Francis Sommerville teaches mathematic and computer courses at E.P. Scarlett High School, Calgary. Sommerville served as an executive member of MCATA and most recently as co-editor of the MCATA newsletter.

Computer programming effectively extends problem solving activities in the mathematics classroom. The computer forces a systematic approach, provides immediate feedback and appeals to most students.

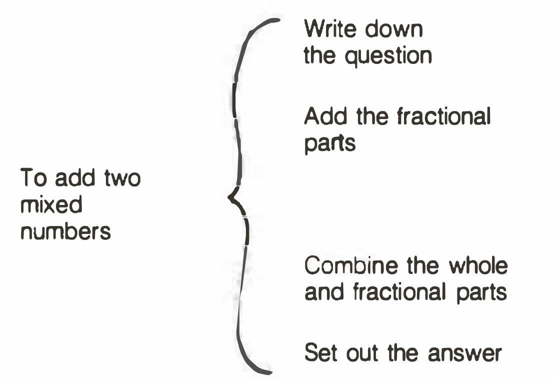
Unfortunately, many programming activities are only marginally related to the responsibilities of junior high math teachers. In his book *Mindstorms: Children, Computers, and Powerful Ideas*, Seymour Papert claims that when children program computers they "establish an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building" (Papert 1980, 5). This sounds great, but will they pass the math final?

An approach is required that allows for group instruction, provides for individual differences and draws examples from the standard mathematics curriculum.

Consider the task of adding mixed numbers. Although not a problem in the usual sense, the task can be approached as a problem solving activity. Some students require extensive review while others are ready to solve the problem in the general sense using variables.

"How do you add mixed numbers?" In response to this question, students might suggest following steps such as writing down the question, adding the fractional part, combining the whole and fraction part and setting out the answer. These responses reflect the usual heuristics of thinking of a similar problem or, dividing the problem into subproblems. These steps could be summarized on the board. (See Figure 1.)

Figure 1. Adding Mixed Numbers

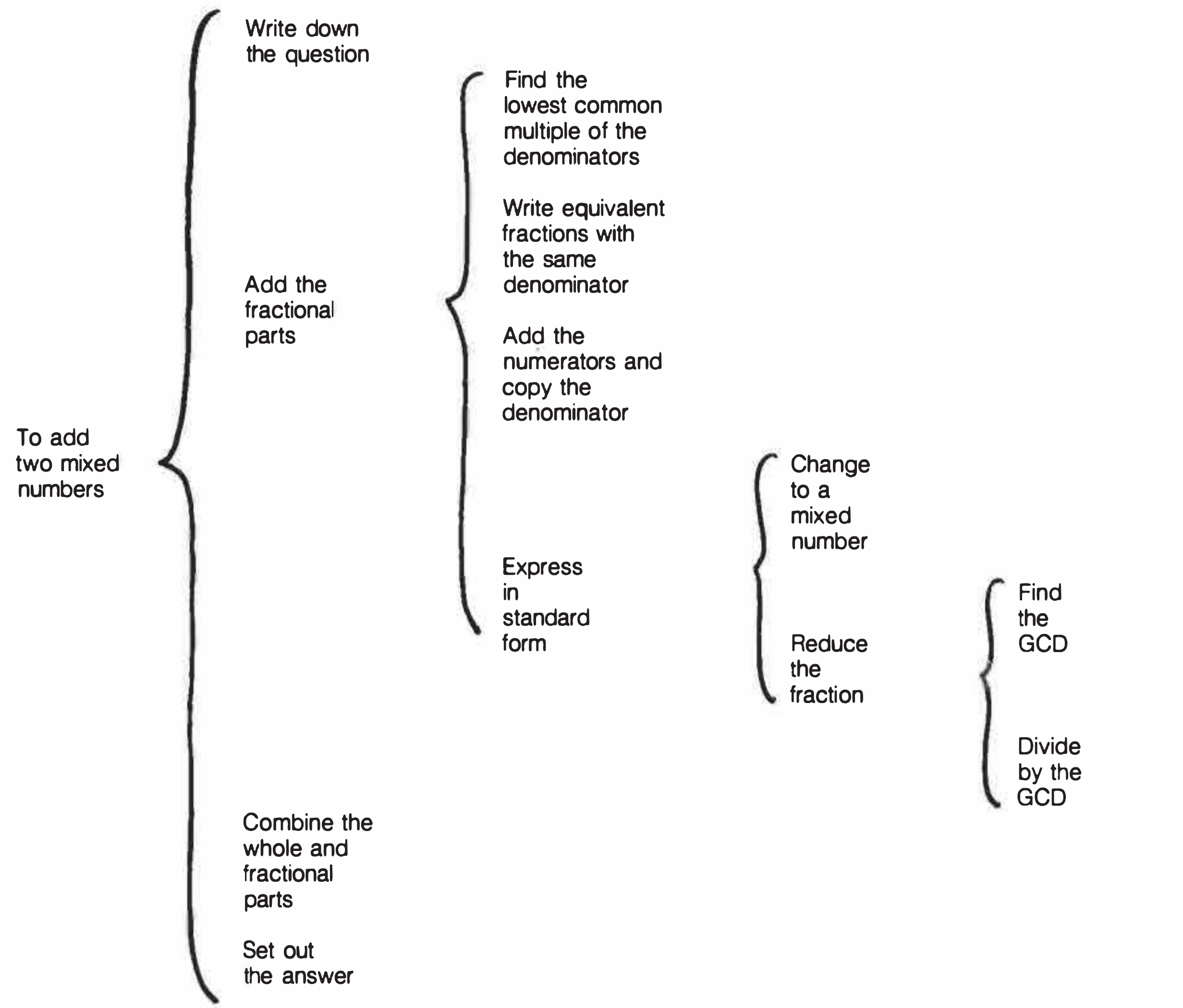


This style of representing the students' suggestions is described by Higgins (1979) and is called a Warnier/Orr diagram. Set notation shows the division of a task into its components. When read from left to right, the diagram explains how to add mixed numbers. The tasks are listed in order from top to bottom. Reading from right to left explains why a certain process is required. Students could be asked to expand on the more difficult aspects of the process to produce a diagram similar to Figure 2.

Figure 2 provides an overall picture and may help students identify areas that require review. As examples are completed, the teacher refers to the stage at which the students are working. This process also provides a sound basis for the student who is ready to solve the problem of adding mixed numbers in the general sense, using algebra.

A major thrust of the junior high mathematics program is to move students from solving specific examples to solving general problems. The general

Figure 2. Overall Picture of Process



solution can be expressed verbally, algebraically or, more recently, as a computer program. Probably all three can be used effectively by some students; therefore a conscious effort to structure examples in a manner that promotes transfer to all three modes of expression is needed. Consider the subproblem identified previously, that of finding the Lowest Common Multiple (LCM) of the denominators.

Once students demonstrate the ability to find the LCM for specific examples, they might be led to a general solution by asking such questions as these (answers are given in parenthesis).

To find the LCM of C and D let us first consider multiples of C. What is the first multiple of C? (C or C times 1.) What is the second multiple of C? (C times 2.) What is the third multiple? (C times 3.) Now, what number is a multiple of both C and D? (C times D.) Is this the lowest common multiple? (Not necessarily.) How could you check to see if there is a smaller multiple of both C and D? (By testing each multiple of C to see if it is a multiple of D.)

This approach transfers to programing in BASIC in the following manner:

Figure 3. Warnier/Orr Solutions

- To add two mixed numbers
  - Write down the question

    ```
    10 INPUT A,E,C,B,F,D
    12 PRINT A" "E"/"C" + "B" "F"/"D" = ";
    ```

  - Add the fractional parts
    - Find the lowest common multiple of the denominators

      ```
      14 FOR X = 1 TO D
      16 LET M = C*X
      18 IF M/D = INT(M/D) THEN 22
      20 NEXT X
      ```

    - Write equivalent fractions with the same denominator

      ```
      22 LET G = M/C*E
      24 LET H = M/D*F
      ```

    - Add the numerators and copy the denominator

      ```
      26 LET N = G+H
      28 LET D = M
      ```

    - Express in standard form
      - Change to a mixed number

        ```
        30 LET W = INT(N/D)
        32 LET N = N-W*D
        ```

      - Reduce the fraction
        - Find the GCD

          ```
          34 FOR F = D TO 1 STEP -1
          36 IF N/F <> INT(N/F) THEN 40
          38 IF D/F = INT (D/F) THEN 42
          40 NEXT F
          ```

        - Divide by the GCD

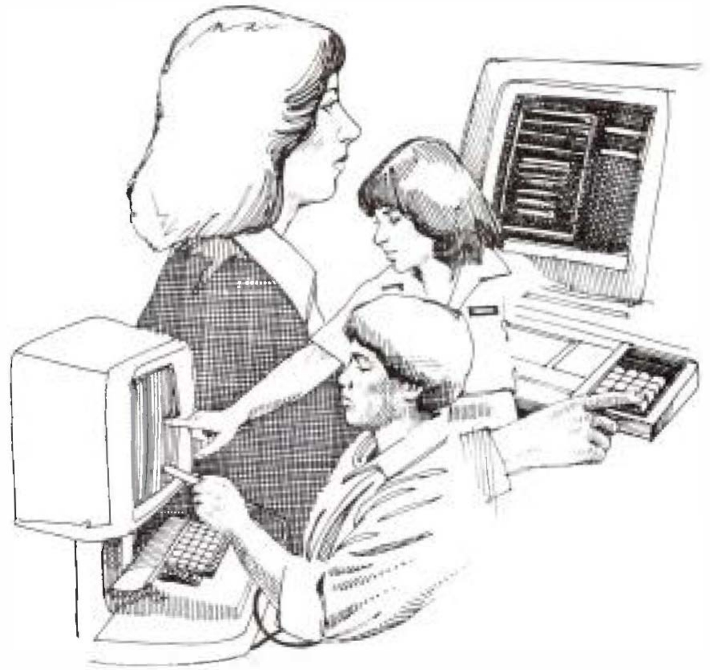
          ```
          42 LET N = N/F
          44 LET D = D/F
          ```

  - Combine the whole and fractional parts

    ```
    46 LET T = A+B+W
    ```

  - Set out the answer

    ```
    48 PRINT T" "N"/"D
    ```



1. FOR × = 1 TO D
2. LET M = C*X : REM M is a multiple of C
3. IF M/D = INT(M/D) THEN 5 : REM Check to see if M is a multiple of D and, if it is, go on to line 5.
4. NEXT × : REM Repeat lines 2 and 3 with the next value of x.
5. PRINT M : REM M is the LCM of C and D.

Marcia Linn (1985) calls such programs "templates." She defines templates as "stereotypic patterns of code using more than a single language feature. They are employed as an entity in programs to perform commonly encountered tasks." In mathematics classes, templates are not necessarily programming code; they could be expressed verbally or algebraically. However, introducing programming code at this level reinforces verbal and algebraic templates.

To complete the initial problem, students must integrate such other templates as finding the greatest common factor and changing improper fractions to mixed numbers. The Warnier/Orr diagram (Figure 3) shows the complete solution. The BASIC program assumes that you are adding the mixed numbers of A E/C and B F/D. (Note that A and B are whole numbers with fractional parts, E/C and F/D respectively.) The program works on Apple or IBM computers. (The input and output commands on lines 10, 12 and 48 are potential problems if this program is attempted on other computers.)

Thus the student solves problems by dividing complex problems into sets of subproblems and solving them by relying on a previously developed repertoire of templates. If programming is part of this process, then students have to learn the features of the language, develop a repertoire of templates and develop the ability to use templates to solve more complex problems.

Clearly programming is not problem solving. It is, however, an exciting subset of the problem solving process as it exits in junior high mathematics classes.

References

Higgins, David A. *Program Design and Construction.* Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

Linn, Marcia C. "The Cognitive Consequences of Programming Instruction in Classrooms." *Educational Researcher,* May 1985, pp. 14–16 and 25–29.

Papert, Seymour. *Mindstorms: Children, Computers and Powerful Ideas.* New York: Basic Books, 1980.